

Automatic Configuration of Performance Management Tools

Technical Field

The technical field is software systems designed to monitor performance of a system.

Background

Performance of modern computer systems, including networked computer servers, may degrade for a variety of reasons, many of which relate to the use of shared resources including disk bandwidth, memory capacity, and central processing unit (CPU) utilization. Information technology (IT) system administrators track performance of their computer systems to ensure optimum allocation of these and other shared resources. Performance monitoring software provides system administrators with the tools necessary to track system performance and to diagnose problems. The performance monitoring software may provide performance information about a computer system, allow the administrator to examine the status of various services hosted by the computer system, identify and resolve bottlenecks, and tune the system and services for more efficient operation. The performance monitoring software may keep a history of the computer system performance, monitoring performance as a background task, and may send alarms for impending performance problems. Using the performance monitoring software, the administrator can pinpoint trends in computer system service levels, and can use this information to balance workloads to accurately plan for computer system growth.

Unfortunately, performance management software is typically complex, and takes a long time to master. Furthermore, performance management software commonly suffers from the inability to adapt to differing environments that the software is expected to monitor. Applications are mixed and configured differently on nearly every server, and server consolidation environments often support several disparate applications and services on a single system. Setup and custom configuration of performance tools specific to these different service environments is a burden on the system administrator, but is required to obtain value. Monitoring software that attempts to avoid this startup overhead suffers from the problem of being too generic to be useful in a modern, complex computing environment.

Summary

To achieve out-of-the-box performance management solutions in modern, complex computer environments, a method and an apparatus provide automatic discovery of applications resident and services active on a monitored computer

1 system. The method and apparatus also provide autonomous, automatic configuration
2 of available performance management tools.

3 Upon initiation of the method, specific information is obtained from the
4 computer system under analysis. First, a discovery process to produce an inventory of
5 installed applications is initiated. Different computing architectures have different
6 mechanisms for obtaining this inventory information. The method and apparatus use
7 the different mechanisms to produce an inventory of all installed applications.

8 Second, an inventory of performance management software is completed. The
9 method and apparatus are flexible enough to provide automatic configuration of
10 different management tools. Thus, the performance management tools present in the
11 computer system must also be determined by a similar process as is used to discover
12 resident applications.

13 Third, an inventory of active services is completed. For the purpose of this
14 method, a service is considered an active instantiation of an application. For example,
15 a database application may be installed on the computer system under analysis, and it
16 might be instantiated in three separate instances, or services, to perform different
17 functions. Likewise, some applications may be installed on a computer system yet
18 not be activated. Configuration of performance tools may be dependent both upon the
19 applications installed on the system, and the nature of how they are activated as
20 services at any given time. Architecture-specific queries of active services,
21 represented by running processes, are used to supplement the internal list of
22 applications and flag some or all of them as services that are enabled or active. The
23 automatically derived inventories of applications, performance tools, and active
24 services that result from these first steps may then be written to an ASCII-format file.

25 Fourth, the inventories serve as input to a step that uses the awareness of the
26 computing environment to generate configuration files specific to the performance
27 management software present on the computer system. The configuration files
28 include collection parameters, application-specific interfaces, performance thresholds,
29 and alarms applicable to specific tools.

30 Fifth, and finally, the performance management software is restarted or
31 reinitialized to engage the new configuration defined by the modified configuration
32 files.

33 The method allows completion of the above-described steps without any
34 manual intervention. For example, performance tool installation may invoke this

1 method during the software install process to automatically configure the performance
2 management tools to the computer system environment. If the computing
3 environment changes, the method may be invoked on demand, or scheduled to
4 execute periodically.

5 While in an embodiment, the method is entirely automatic; the automatic
6 method may be supplemented manually by amending the output of the first three steps
7 and then starting the process from the fourth step. This manual intervention can be
8 provided without the requirement for manual setup and configuration.

9 **Description of the Drawings**

10 The detailed description will refer to the following drawings in which like
11 numerals refer to like items, and in which:

12 Figure 1 is a block diagram of an apparatus that provides for automatic
13 configuration of performance management tools; and

14 Figure 2 is a flow chart showing a method for automatic configuration of
15 performance management tools.

16 **Detailed Description**

17 Figure 1 is a block diagram of an exemplary apparatus that can be used for
18 automatic configuration of performance management tools.

19 A computer system 100 environment includes operating or control systems
20 (OS) 101 supplemented by applications 102 resident on storage media 103, which
21 may include disk or networked storage. The computer system 100 has interfaces that
22 allow specialized programs to discover installed applications. For example, using
23 Windows NT, the OS 101 includes software to query a database of installed
24 applications. This information includes registry information for installed performance
25 management tools as well. On hp-ux, a Software Distributor tool may be used in an
26 analogous fashion to derive an inventory of installed applications. Similar
27 functionality exists in conjunction with all major Operating Systems. To augment
28 this, programs may traverse mounted file systems and directories on disk and network
29 media to determine which programs and applications are available to be executed on
30 the system. When applications are added to the environment, an inventory is updated
31 to reflect the addition. Thus, in an embodiment, a discovery process or module 110
32 can query a registration database or scan storage media to discover installed
33 applications and to construct an ASCII (human-readable) inventory file 120 listing
34 applications 121. This same procedure can be executed specifically to add to a

1 performance management tool inventory 122 all of the performance management
2 tools that are installed on the system.

3 During system operation, a varying number of installed applications 121 will
4 be active at any one time. Executing applications compose services that are the
5 function of a computer system. Services are composed of one or more processes
6 executing programs specific to a given application. Some applications may be
7 instantiated several different times to perform different services. In any case,
8 computer system environments have interfaces (such as perflib on Windows NT, pstat
9 on hp-ux) allow software to query the execution environment to discover which
10 applications are currently active. Thus, in an embodiment, the discovery process 110
11 can supplement the inventory file 120 with information regarding active applications,
12 or the current set of services instantiated 123.

13 The inventory file 120 may contain entries for each of the installed
14 applications 121, the performance management tools 122, and the active services 123.
15 Alternatively, the inventory file 120 may include an entry for all applications, and
16 flags may be set to indicate an active application and performance tools. The
17 discovery process 110 functions to maintain the inventory file 120 up-to-date. Thus, a
18 complete inventory file 120 includes all possible services and performance tools in the
19 environment along with information as to which services and tools are active.

20 The inventory file 120 thus generated may be edited by a system
21 administrator. This step is allowed because the discovery process 110 may not be
22 comprehensive, and the human-readable nature of the inventory allows for manual
23 intervention based on the administrator's knowledge. In a current state of the art,
24 system administrators need to manually configure performance tool configuration
25 files directly, sometimes repeatedly and differently for every installed application and
26 service. Allowing the administrator to embed this knowledge into a stable inventory
27 allows for automation of much of this process.

28 A software module, or process, executing on the computer system may use the
29 resulting inventories of installed applications 121, performance tools 122, and
30 instantiated services 123 to generate configuration files specific to the computing
31 environment. In an embodiment, a file generation process 130 uses the inventory file
32 120 to generate performance tool configuration files 140 specific to the performance
33 management tools that are present on the system 100. The configuration files 140
34 thus generated may include collection parameters 141, application-specific interfaces

1 142, and performance thresholds/alarms 143 applicable to specific performance
2 management tools.

3 Figure 2 is a flowchart showing one method according to the apparatus shown
4 in Figure 1. A process 200 is executed on the computer system 100, and begins at
5 block 205, when the computer system 100 is started, or at some other time. In block
6 210, installed applications 121 are discovered by reading the system registry or by
7 scanning attached storage devices on the computer system 100. In block 215, the
8 same or a similar discovery process is used to add all installed performance
9 management tools 122 to the inventory file 120. In block 220, the active process
10 environment is examined to augment the inventory file 120 with active service 123
11 information. In an embodiment, the discovery processes 110 may be executed only
12 periodically as new applications are added to the system 100, while the active service
13 discovery may be executed on a frequent basis as the mix of running processes
14 change. The augmented inventory file 120 thus created is maintained until updated
15 by a subsequent inventory.

16 The inventory file 120, as an ASCII-format file, is available for viewing by an
17 operator or administrator of the system, and may be printed. In block 225, the
18 inventory file 120 itself may be edited by the administrator to change or add
19 information relevant to the environment. Manual intervention is not required,
20 however.

21 In block 230, the configuration file generation process 130 uses the inventory
22 file 120 to create configuration files 140 specific to the performance management
23 tools shown in the inventory steps (210, 215, 220). The configuration files 140
24 typically vary depending on particular performance management tool, but may
25 include collection parameters, application-specific interfaces, performance thresholds,
26 and alarms specific to the tool as well as the current operating service environment.

27 In block 235, the performance management software is reinitialized or
28 restarted to engage the new configuration shown in the configuration files 140.
29 Restarting the performance management software may be accomplished
30 automatically, without intervention of the administrator, or would occur the next time
31 the computer system is restarted. If the computing environment changes (for
32 example, a new application is installed), the entire process can be re-iterated (block
33 240) in order to reflect the changes in the configuration files. If the process does not
34 need to be repeated, the process then ends with block 245.

1 The above-described process is but one method for automatic configuration of
2 performance management tools, and other methods may be used with the system
3 shown in Figure 1. The method illustrated in Figure 2 may initiate automatically on
4 start up of the system. The methods may also be initiated on demand by the system
5 administrator, for example, when new hardware devices or application programs are
6 added. The methods may also be initiated periodically to maintain optimum
7 configuration of the performance management tools as the computer system evolves.
8 For example, the methods may be initiated once per week, or more frequently.